Differentially Private Publication of Social Graphs at Linear Cost

Hiep H. Nguyen, Abdessamad Imine, and Michaël Rusinowitch
LORIA/INRIA Nancy-Grand Est, France
Email: {huu-hiep.nguyen,michael.rusinowitch}@inria.fr, abdessamad.imine@loria.fr

***Abstract*—The problem of private publication of graph data has attracted a lot of attention recently. The prevalence of differential privacy makes the problem more promising. However, a large body of existing works on differentially private release of graphs have not answered the question about the upper bounds of privacy budgets. In this paper, for the first time, such a bound is provided. We prove that with a privacy budget of $O(\log n)$, there exists an algorithm capable of releasing a noisy output graph with edge edit distance of $O(1)$ against the true graph. At the same time, the complexity of our algorithm Top-m Filter is linear in the number of edges m. This lifts the limits of the state-of-the-art, which incur a complexity of $O(n^2)$ where n is the number of nodes and runnable only on graphs having n of tens of thousands.**

I. Introduction

As one of the most general forms of data representation, graphs support all aspects of the relational data mining process. With the emergence of increasingly complex networks [10], the research community requires large and reliable graph data to conduct in-depth studies. However, this requirement usually conflicts with privacy policies of data contributing entities. Naive approaches like removing user ids from a social graph are not effective, leaving users open to privacy risks, especially re-identification attacks [1] [7].

In this paper, we address the problem of graph anonymization from the perspective of differential privacy. This privacy model offers a formal definition of privacy with a lot of interesting properties: no computational/informational assumptions about attackers, data type-agnosticity, composability and so on [9]. By differential privacy, we want to ensure the existence of connections between users to be hidden in the released graph while retaining important structural information for graph analysis [11], [12], [13], [3], [14].

Differentially private algorithms relate the amount of noise to the sensitivity of computation. Lower sensitivity implies smaller added noise. Because edges in simple undirected graphs are usually assumed independent, standard Laplace mechanism is applicable (e.g. adding Laplace noise to each cell of the adjacency matrix). However, this approach may severely deteriorate graph structure. Recent methods of graph release under differential privacy try to reduce the graph sensitivity by many ways. Schemes in [11], [12] use *dK-series*[8] to summarize the graph into a distribution of degree correlations. The global sensitivity of 1K-series (resp. 2K-series) is 4 (resp. $O(n)$). Lower sensitivity of $O(\sqrt{n})$ is proposed in [13] by graph spectral analysis. The most recent works [3], [14] even reduce the sensitivity of graph to $O(\log n)$. While *Density Explore Reconstruct* (DER)[3] employs a data-dependent quadtree to summarize the adjacency matrix into a counting tree, Xiao et al. [14] propose to use *Hierarchical Random Graph* (HRG) [4] to encode graph structural information in terms of edge probabilities. A common disadvantage of the state-of-the-art DER [3] and HRG-MCMC [14] is the scalability issue. Both of them incur quadratic complexity $O(n^2)$, limiting themselves to medium-sized graphs.

To remedy the scalability problem, we propose *Top-m Filter* (TmF) algorithm, which runs in $O(m)$, linear in the number of edges. By considering the adjacency matrix as a sparse dataset, TmF leverages the high-pass filtering technique in [5] to avoid the whole matrix manipulation. More importantly, via TmF, we provide a theoretical result stating that $O(\log n)$ is an upper bound for graph release under differential privacy. This naturally rules out high-sensitivity schemes in [11], [12], [13] and makes DER, HRG-MCMC meaningful only in regimes of small privacy budgets (i.e. not exceeding $O(\log n)$).

II. Preliminaries

In this section, we review key concepts and mechanisms of differential privacy.

A. Differential Privacy

Essentially, ϵ-differential privacy (ϵ-DP) [6] is proposed to quantify the notion of *indistinguishability* of neighboring databases. In the context of graph release, two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are neighbors if $V_1 = V_2$, $E_1 \subset E_2$ and $|E_2| = |E_1| + 1$. Formal definition of ϵ-DP for graph data is as follows.

Definition 2.1: A mechanism $\mathcal{A}$ is ϵ-differentially private if for any two neighboring graphs G_1 and G_2, and for any output $O \in Range(\mathcal{A})$,

$$Pr[\mathcal{A}(G_1) \in O] \leq e^{\epsilon} Pr[\mathcal{A}(G_2) \in O]$$

Laplace mechanism [6] and Exponential mechanism [9] are two standard techniques in differential privacy. The latter is a generalization of the former. Laplace mechanism is based on the concept of *global sensitivity* of a function f which is defined as $\Delta f = \max_{G_1,G_2} ||f(G_1) - f(G_2)||_1$ where the maximum is taken over all pairs of neighboring G_1, G_2. Given a function f and a privacy budget ϵ, the noise is drawn from a Laplace distribution $p(x|\lambda) = \frac{1}{2\lambda} e^{-|x|/\lambda}$ where $\lambda = \Delta f/\epsilon$.

Theorem 2.1: (Laplace mechanism [6]) For any function $f : G \to \mathbb{R}^d$, the mechanism $\mathcal{A}$

$$\mathcal{A}(G) = f(G) + \langle Lap_1(\frac{\Delta f}{\epsilon}), ..., Lap_d(\frac{\Delta f}{\epsilon})\rangle \quad (1)$$

satisfies ϵ-differential privacy, where $Lap_i(\frac{\Delta f}{\epsilon})$ are i.i.d Laplace variables with scale parameter $\frac{\Delta f}{\epsilon}$. □

Composability is a nice property of differential privacy which is not satisfied by other privacy models such as k-anonymity. Specifically, serial and parallel compositions are key ingredients in our algorithm TmF (Section III).

Theorem 2.2: (Sequential and parallel compositions [9]) Let each A_i provide ϵ_i-differential privacy. A sequence of $A_i(D)$ over the dataset D provides $\Sigma_{i=1}^n \epsilon_i$-differential privacy.

Let each A_i provide ϵ_i-differential privacy. Let D_i be arbitrary disjoint subsets of the input domain D. The sequence of $A_i(D_i)$ provides $\max_{i=1}^n \epsilon_i$-differential privacy. □

III. Top-m Filter

We introduce our linear time algorithm *Top-m Filter* (TmF) in this section by considering the adjacency matrix as a sparse contingency table. TmF uses an idea similar to High-pass Filter in [5] to avoid the materialization of the noisy adjacency matrix. Our algorithm is therefore linear in the number of edges. By devising TmF, we also reach an upper bound on privacy budget for graph publication in ϵ-DP setting.

A. Overview

Given the input graph G (represented by an adjacency matrix A) and privacy budget ϵ, by the assumption of edge independence, the naive approach (*Naive*) adds Laplace noise to all cells in the upper-triangle of A, i.e. $\tilde{A}_{ij} = A_{ij} + Lap(1/\epsilon)$ for all $j > i \geq 1$. $\tilde{A}_{ij}$ is then post-processed by rounding $\hat{A}_{ij} = \arg\min_{x=0,1} |\tilde{A}_{ij} - x|$.

Instead of processing each cell independently as in Naive approach, our idea is to keep top-m noisy values $\tilde{A}_{ij}$ and reconstruct them to 1-cells. However, the number of edges m needs to be first obfuscated (note that in edge privacy model, only n is public [14]). We can achieve this by *Laborious filtering*, i.e. first computing the noisy number of edge $\tilde{m} = m + Lap(1/\epsilon_2)$, then adding Laplace noise $Lap(1/\epsilon_1)$ to all $\frac{n(n-1)}{2}$ cells and selecting top-$\tilde{m}$ noisy cells. This approach costs $O(n^2)$ in space and $O(n^2 \log n)$ in time because of the materialization of all cells. TmF avoids such problem by computing the threshold θ so that there are exactly $\tilde{m}$ noisy cells larger than θ. We call those cells *passing cells*. Fig. 1 depicts the processes of Naive, Laborious filtering and TmF.

We have two cases: $0 < \theta < 1$ and $1 \leq \theta$. The case $\theta \leq 0$ results in the number of passing cells is at least $\frac{n(n-1)}{4} \gg \tilde{m}$, so omitted.

Case 1: $0 < \theta < 1$: the number of passing 1-cells is

$$n_1 = m \int_\theta^{+\infty} \frac{\epsilon_1}{2} \exp(-\epsilon_1|x-1|)dx = \frac{m}{2}(2 - e^{-\epsilon_1(1-\theta)}) \quad (2)$$

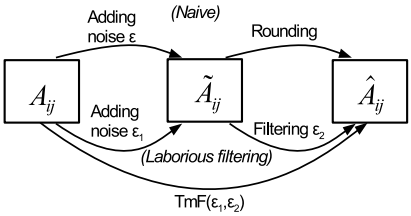


Fig. 1: TmF algorithm

The number of passing 0-cells is

$$n_0 = (\frac{n(n-1)}{2} - m) \int_\theta^{+\infty} \frac{\epsilon_1}{2} \exp(-\epsilon_1|x|)dx$$
$$= (\frac{n(n-1)}{2} - m)\frac{1}{2}e^{-\epsilon_1\theta}$$

By equating the sum of n_1 and n_0 to $\tilde{m}$, we can compute the value of θ. Because $\tilde{m} = m + Lap(1/\epsilon_2)$, we have $E[\tilde{m}] = m$. So to simplify the calculations, we set $n_1 + n_0 = m$. This leads to

$$\theta = \frac{1}{2\epsilon_1} \ln(\frac{n(n-1)}{2m} - 1) + \frac{1}{2} \quad (3)$$

Case 2: $1 \leq \theta$: Similarly, the value of θ is

$$\theta = \frac{1}{\epsilon_1} \ln(\frac{n(n-1)}{4m} + \frac{1}{2}(e_1^\epsilon - 1)) \quad (4)$$

In Algorithm 1, we replace all m by $\tilde{m}$.

B. Algorithm

To decide whether $\theta \geq 1$ or $0 \leq \theta \leq 1$, we compute the threshold ϵ_t of ϵ_1 at $\theta = 1$. For both cases,

$$\theta = 1 \leftrightarrow \epsilon_t = \ln(\frac{n(n-1)}{2m} - 1) \quad (5)$$

Theorem 3.1: The complexity of TmF is $O(m)$ □

Proof: Processing 1-cells (Lines 10-15) runs in $O(m)$. The maximum value of n_0 (Line 17) is $\tilde{m}$ ($= m$ in expectation). For each 0-cell to be processed, the rejection sampling (Line 19) succeeds with probability at least $1 - \frac{2m}{n(n-1)} = 1 - O(1/n)$. So in summary, the total complexity of TmF is $O(m)$. ■

Theorem 3.1 makes sense if we consider the complexity $O(n^2)$ of the state-of-the-art DER [3] and HRG-MCMC [14].

C. Privacy Analysis

In this section, we show that TmF satisfies ϵ-DP where $\epsilon = \epsilon_1 + \epsilon_2$. Our TmF consists of two steps. It is easy to verify that the sensitivity of m is 1. The first step of computing $\tilde{m}$ satisfies ϵ_2-DP. The second step of processing 1-cells and 0-cells is equivalent to independently adding noise $Lap(1/\epsilon_1)$ to each cell and letting them go through a high-pass filter with threshold θ. The sensitivity of each cell is also 1. By the assumption of edge independence, parallel composition (Theorem 2.2) is applicable at cell level. So the second step satisfies ϵ_1-DP. By sequential composition (Theorem 2.2), TmF satisfies $(\epsilon_1 + \epsilon_2)$-DP as stated in the following theorem.

Algorithm 1 Top-m Filter

Input: input graph $G = (V, E)$, privacy parameters ϵ_1, ϵ_2
Output: sanitized graph $\tilde{G}$

$\tilde{G} \leftarrow \varnothing$
// compute $\tilde{m}$ and θ
$\tilde{m} = m + Lap(1/\epsilon_2)$
$\epsilon_t = \ln(\frac{n(n-1)}{2\tilde{m}} - 1)$
if $\epsilon_1 < \epsilon_t$ **then**
    $\theta = \frac{1}{2\epsilon_1} \ln(\frac{n(n-1)}{2\tilde{m}} - 1)$
else
    $\theta = \frac{1}{\epsilon_1} \ln(\frac{n(n-1)}{4\tilde{m}} + \frac{1}{2}(e^{\epsilon}_1 - 1))$
// process 1-cells
$n_1 = 0$
for $A_{ij} = 1$ **do**
    compute $\tilde{A}_{ij} = A_{ij} + Lap(1/\epsilon_1)$
    if $\tilde{A}_{ij} > \theta$ **then**
        add edge (i, j) to $\tilde{G}$
        n_1 ++
// process 0-cells
$n_0 = \tilde{m} - n_1$
while $n_0 > 0$ **do**
    random pick an edge (i, j)
    if $\tilde{G}$ does not contain (i, j) **then**
        add edge (i, j) to $\tilde{G}$
        n_0- -
return $\tilde{G}$

Theorem 3.2: TmF satisfies ϵ-DP where $\epsilon = \epsilon_1 + \epsilon_2$. □

Now we proceed to the more important result: TmF could reduce the edit distance between $\tilde{G}$ and G to $O(1)$ at $\epsilon_1 = O(\log n)$. The edit distance is defined as

$$D(G, \tilde{G}) = \frac{1}{2}(|E_G \setminus E_{\tilde{G}}| + |E_{\tilde{G}} \setminus E_G|) \tag{6}$$

By the analysis in section III-A, the expected number of passing 1-cells is n_1, so the expected edit distance $D(G, \tilde{G}) = m - n_1$. At $\theta = 1$, we have $n_1 = \frac{m}{2} = D(G, \tilde{G})$ and $\epsilon_1 = \epsilon_t = \ln(\frac{n(n-1)}{2\tilde{m}} - 1)$. The cases of small edit distance therefore correspond to the case $0 < \theta < 1$. Setting $D(G, \tilde{G}) = \gamma m$, $\gamma \in [\frac{1}{m}, 1]$, we need to find the value of ϵ_1.

$$\begin{aligned}
& D(G, \tilde{G}) = \gamma m \\
\leftrightarrow \quad & m - \frac{m}{2}(2 - e^{-\epsilon_1(1-\theta)}) = \gamma m \\
\leftrightarrow \quad & e^{-\epsilon_1(1-\theta)} = 2\gamma \\
\leftrightarrow \quad & \theta = 1 + \frac{1}{\epsilon_1} \ln(2\gamma) \\
\leftrightarrow \quad & \frac{1}{2\epsilon_1} \ln(\frac{n(n-1)}{2m} - 1) + \frac{1}{2} = 1 + \frac{1}{\epsilon_1} \ln(2\gamma) \quad \text{(from (3))} \\
\leftrightarrow \quad & \epsilon_1 = \ln(\frac{\frac{n(n-1)}{2m} - 1}{4\gamma^2})
\end{aligned}$$

Because real-world graphs are usually sparse, $m = O(n)$, we reach $\epsilon_1 = O(\log n)$. Specifically, $\epsilon_1 \approx 3 \ln n$, $\epsilon_1 \approx 2 \ln n$, $\epsilon_1 \approx \ln n$ at $\gamma = \frac{1}{m}$, $\gamma = \frac{1}{\sqrt{m}}$ and $\gamma = \frac{0.5}{O(\sqrt{d})}$ respectively ($\bar{d} = \frac{2m}{n}$ is the average degree). We come up with the following theorem.

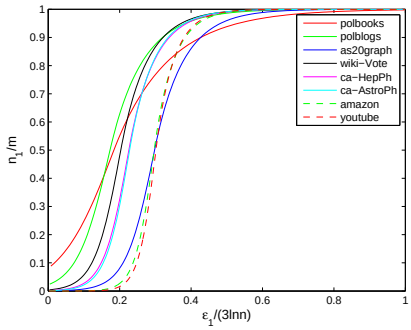


Fig. 2: n_1/m as a function of $\epsilon_1/(3 \ln n)$

Theorem 3.3: TmF can make the edit distance $D(G, \tilde{G}) = O(1)$ at $\epsilon_1 \approx 3 \ln n$. □

Fig. 2 shows the normalized number of passing 1-cells n_1/m as a function of $\epsilon_1/(3 \ln n)$ over eight datasets (cf. Table I). As we can see, at $\epsilon_1 = \ln n$, 65-90% of the edges in G are kept in $\tilde{G}$.

This result naturally points out the waste of privacy budget in [11], [12] and [13] where $\epsilon = O(\sqrt{n})$ or $\epsilon = O(n)$. Interestingly, in HRG-MCMC scheme [14], the sensitivity $\Delta u \approx 2 \ln n$ which means the scheme corresponds to our case $\gamma = \frac{1}{\sqrt{m}}$ if we set $\epsilon_1 = \Delta u$.

IV. Evaluation

In this section, our evaluation aims to show the efficiency (in runtime) of TmF and compare its effectiveness (in terms of utility metrics) with HRG-MCMC and DER. We pick six small and medium-sized graphs and two large ones [1]. In Table I, logLK_1 and logLK_2 are the log-likelihoods of the dendrograms [14] created by Louvain algorithm [2] and bottom-up binary construction (i.e. nodes 1 and 2 are paired, nodes 3 and 4 are paired and so on) respectively.

A. Utility Metrics

We use the following statistics for utility measurement
- Average degree: $S_{AD} = \frac{1}{n} \sum_{v \in V} d_v$
- Maximal degree: $S_{MD} = \max_{v \in V} d_v$
- Degree variance: $S_{DV} = \frac{1}{n} \sum_{v \in V} (d_v - S_{AD})^2$
- Power-law exponent of degree sequence: S_{PL} is the estimate of γ assuming the degree sequence follows a power-law $\Delta(d) \sim d^{-\gamma}$
- Average distance: S_{APD} is the average distance among all pairs of vertices that are path-connected.
- Effective diameter: S_{EDiam} is the 90-th percentile distance among all path-connected pairs of vertices.
- Connectivity length: S_{CL} is defined as the harmonic mean of all pairwise distances in the graph.
- Diameter : S_{Diam} is the maximum distance among all path-connected pairs of vertices.
- Clustering coefficient: $S_{CC} = \frac{3N_\Delta}{N_3}$ where N_Δ is the number of triangles and N_3 is the number of connected triples.

[1] available at http://www-personal.umich.edu/~mejn/netdata/ and http://snap.stanford.edu/data/index.html

- Degree distribution: S_{DD} is the normalized degree histogram.

- Distance distribution: S_{PDD} is the normalized node-pair shortest-path histogram.

All of the above statistics are taken average over 10 sample graphs. $S_{APD}, S_{CL}, S_{EDiam}, S_{Diam}$ are computed exactly in six small graphs. In *amazon* and *youtube*, S_{Diam} is lower bounded by the longest distance among all-destination breadth-first-searches from 1,000 randomly chosen nodes.

The relative error (rel.err) for each metric S is computed as $\frac{|S(G)-S_{avg}(\tilde{G})|}{S(G)}$ except S_{DD} and S_{PDD} whose errors are computed as $|S(G) - S_{avg}(\tilde{G})|_1/2$.

TABLE I: Graph dataset statistics (k:thousand, m:million)

Dataset	#Nodes	#Edges	logLK$_1$	logLK$_2$
polbooks	105	441	-950	-1248
polblogs	1,124	16,715	-66k	-74k
as20graph	6,474	12,572	-71k	-100k
wiki-Vote	7,115	100,762	-576k	-618k
ca-HepPh	12,006	118,489	-383k	-876k
ca-AstroPh	18,771	198,050	-904k	-1.54m
amazon	334k	925k	-2.99m	-2.88m
youtube	1,134k	2,987k	-18.02m	-11.14m

B. Effectiveness of TmF

We assess the utility of TmF by varying ϵ_1 (Fig. 3) while fixing $\epsilon_2 = 1.0$. As ϵ_1 increases (lower privacy guarantee), we gain better utility (lower relative errors). For six small graphs, TmF utility scores are nearly linear in ϵ_1 for ϵ_1 in the range $[0, \ln n]$. As ϵ_1 exceeds the threshold ϵ_t (Fig. 2), the edit distance $D(G, \tilde{G})$ decreases quickly, so does the relative error.

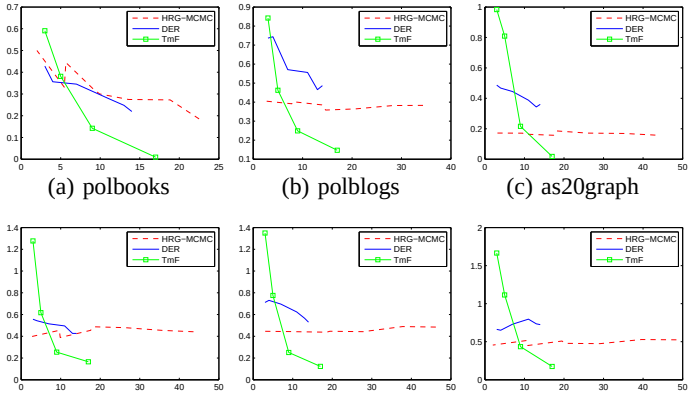
C. Comparative Evaluation

We report comparisons between TmF and HRG-MCMC, DER in Fig. 3. For HRG-MCMC, we vary ϵ_1 between $2\Delta u$ and 1.0 (used in [14]). Clearly, HRG-MCMC and DER also provide better utility at higher $\Sigma\epsilon$. However, the relative error does not change much within a wide range of ϵ. We could explain this phenomenon by the usage of reconstruction in HRG-MCMC and DER. Reconstruction steps make the edit distance $D(G, \tilde{G})$ be $O(m)$. In contrast, TmF, by relating ϵ_1 to the edit distance, makes the relationship between ϵ_1 and relative error much more correlated. Roughly speaking, HRG-MCMC performs best in stringent regime ($\epsilon_1 = 2.0$) whereas TmF outperforms the others at high privacy budgets ($\epsilon_1 = 8.0, 16.0$).

On the runtime, TmF produces a sample graph in less then 10s for *youtube* graph, whereas HRG-MCMC takes one day and DER takes 250s for *ca-AstroPh* graph.

V. Conclusion

We provide an upper bound for privacy budget ϵ that any differentially private scheme for graph release should not exceed. Based on filtering technique, we design the algorithm TmF that reduces the edit distance between the noisy graph and the true graph to $O(1)$. Our scheme TmF can run on million-scale graphs. The comprehensive experiments demonstrate the efficiency and effectiveness of our scheme and explain the loss of information in the state-of-the-art HRG-MCMC and DER. For future work, we intend to (1) include the degree sequence (1K-series [8]) into the scheme to improve the utility, (2) investigate other summary structures for graphs other than HRG.



(a) polbooks (b) polblogs (c) as20graph

(d) wikiVote (e) caHepPh (f) caAstroPh

Fig. 3: Comparison of utility in terms of relative error (Y-axis) vs. ϵ (X-axis)

References

[1] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW*, pages 181–190. ACM, 2007.

[2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[3] R. Chen, B. C. Fung, P. S. Yu, and B. C. Desai. Correlated network data publication via differential privacy. *VLDB Journal*, 23(4):653–676, 2014.

[4] A. Clauset, C. Moore, and M. E. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

[5] G. Cormode, C. Procopiuc, D. Srivastava, and T. T. Tran. Differentially private summaries for sparse data. In *ICDT*, pages 299–311. ACM, 2012.

[6] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *TCC*, pages 265–284, 2006.

[7] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. *VLDB Endowment*, 2008.

[8] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat. Systematic topology analysis and generation using degree correlations. In *SIGCOMM*. ACM, 2006.

[9] F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *SIGMOD*, pages 19–30. ACM, 2009.

[10] M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[11] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao. Sharing graphs using differentially private graph models. In *SIGCOMM*, pages 81–98. ACM, 2011.

[12] Y. Wang and X. Wu. Preserving differential privacy in degree-correlation based graph generation. *TDP*, 6(2):127, 2013.

[13] Y. Wang, X. Wu, and L. Wu. Differential privacy preserving spectral graph analysis. In *PAKDD*. Springer, 2013.

[14] Q. Xiao, R. Chen, and K.-L. Tan. Differentially private network data release via structural inference. In *KDD*, pages 911–920. ACM, 2014.